Detecting network communities by propagating labels under constraints

Michael J. Barber
*AIT Austrian Institute of Technology GmbH, Foresight & Policy Development Department, Vienna, Austria**

John W. Clark
Department of Physics, Washington University, Saint Louis, MO
(Dated: June 18, 2009)

We investigate the recently proposed label-propagation algorithm (LPA) for identifying network communities. We reformulate the LPA as an equivalent optimization problem, giving an objective function whose maxima correspond to community solutions. By considering properties of the objective function, we identify conceptual and practical drawbacks of the label propagation approach, most importantly the disparity between increasing the value of the objective function and improving the quality of communities found. To address the drawbacks, we modify the objective function in the optimization problem, producing a variety of algorithms that propagate labels subject to constraints; of particular interest is a variant that maximizes the modularity measure of community quality. Performance properties and implementation details of the proposed algorithms are discussed. Bipartite as well as unipartite networks are considered.

PACS numbers: 89.75.Hc

I. INTRODUCTION

There is great current interest in identifying communities in networks. Informally, communities in networks, or graphs, are subgraphs whose vertices are more strongly connected to one another than to the vertices outside the subgraph. A variety of approaches have been taken to make concrete the idea of communities, giving rise to a number of efficient methods for community identification (for useful overviews, see Refs. [1, 2, 3]).

Recently, Raghavan et al. [4] have introduced a label-propagation algorithm (LPA) for identifying network communities. Initially, each vertex in the graph is assigned a unique numeric label. The label for each vertex is replaced with the most frequent label from its neighbors. Relabeling continues until a stable set of labels is reached. Network communities are defined as the sets of vertices bearing the same labels. The LPA offers a number of desirable qualities, including conceptual simplicity, ease of implementation, and practical efficiency—the algorithm rapidly [4, 5] finds community assignments of high quality, as measured by the popular modularity measure [6].

The LPA was originally presented operationally, with communities defined as the outcome of a specific procedure. In this work, we consider an equivalent mathematical formulation, in which community solutions are understood in terms of optima of an objective function. We define an objective function H based on the number of edges that connect vertices with identical labels, and show that the LPA identifies local optima of H. This is formally equivalent to minimizing the Hamiltonian for a ferromagnetic Potts model [7]. The mathematical formulation exposes a number of interesting properties of the LPA. A feature of conspicuous importance is that the globally optimal solution for any network is the uninteresting trivial solution in which all vertices are assigned the same label, with other solutions found by label propagation corresponding to suboptimal local maxima of H.

The objective function optimized by label propagation thus corresponds poorly to our conceptual understanding of communities—an increase in H need not produce what we would consider to be better communities. In particular, attempts to improve on the label propagation algorithm by facilitating its escape from local maxima in H may be counterproductive. We demonstrate that this can create practical difficulties for improvement upon the standard LPA.

We next consider adding a term to the original objective function that penalizes undesirable solutions, producing algorithms that propagate labels subject to constraints. We examine several possibilities for the penalty term. Of special interest is a penalty term that works to divide vertices into groups of equal total degree, yielding a label propagation variant that strictly maximizes the modularity [6] while maintaining the favorable computational complexity of the standard LPA. We characterize the effectiveness of the several label propagation algorithms through application to a model network and a selection of real-world networks.

*Electronic address: michael.barber@ait.ac.at

The structure of the remainder of the paper is as follows. In section II we briefly summarize the original operational presentation of the label propagation algorithm [4]. In section III, we reformulate the label propagation algorithm as a mathematical optimization problem, and in section IV consider drawbacks of the LPA thus revealed. We address the drawbacks in section V by adding constraints to the optimization problem, with attendant notes on implementation in the appendices. Performance of several label propagation variants are compared in section VI, for both unipartite and bipartite networks. We conclude with a summary and discussion in section VII.

II. THE LABEL PROPAGATION ALGORITHM

The identification of communities in networks is a topic of great recent interest. Formulation of the problem presents two main challenges. First, the notion of community is imprecise, requiring a definition to be provided for what constitutes a community. Second, community solutions must also be practically realizable for networks of interest. The interplay between these challenges allows a variety of community definitions and community identification algorithms suited to networks of different sizes, as measured by the number of vertices n or edges m in the network.

A prominent formulation of the community-identification problem is based on the modularity Q introduced by Newman and Girvan [6]. The quality of communities given by a partition of the network vertices is assessed by comparing the number of edges between vertices in the same community to the number expected from a null model network. Formally, this is

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - P_{ij} \right) \delta \left(g_i, g_j \right) \quad , \tag{1}$$

where the A_{ij} are components of the adjacency matrix for the network, and g_i is the community for vertex i. The presence of the Kronecker delta term $\delta \left(g_i, g_j \right)$ restricts the sum to edges within communities. The probability of an edge existing between vertices i and j in the null model network is given by P_{ij}. The standard choice of null model takes the probability of an edge to be proportional to the product of the degrees k_i and k_j of the vertices, giving

$$P_{ij} = \frac{k_i k_j}{2m} \quad . \tag{2}$$

With this choice for P_{ij}, the modularity becomes

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta \left(g_i, g_j \right) \quad . \tag{3}$$

Communities are then sought by finding partitions of the set of vertices that have a high value for modularity. The global maximum of Q is generally inaccessible, as the number of possible partitions for a set grows too rapidly to be feasibly examined for all but the smallest networks, although effective heuristics exist for finding high modularity solutions. A seminal example is the greedy agglomerative hierarchical algorithm [8, 9], wherein pairs of communities are successively merged so as to cause the largest possible increase in Q at each step.

Recently, Raghavan et al. [4] have introduced a label-propagation algorithm (LPA) for identifying network communities. In contrast to the above modularity-based approach, communities are defined in the LPA as vertex partitions identified by a specific algorithm. The algorithm is conceptually simple in its operation. Initially, each vertex in the graph is assigned a unique numeric label. The label for each vertex is then replaced with the most frequent label amongst its neighbors; when several labels are equally frequent, the current label is kept if it is among the most frequent, while otherwise a new label is chosen at random from the most frequent. Vertices are repeatedly relabeled, with the algorithm terminating when the label for each vertex is (one of) the most frequent of the labels for the neighbors of the vertex. To avoid possible cycles and ensure termination, Raghavan et al. [4] suggest updating the vertex labels asynchronously and in random order. Network communities are then associated with sets of vertices bearing the same labels.

The LPA offers a number of desirable qualities. As described above, it is conceptually simple, being readily understood and quickly implemented. Communities found can be of high quality, as assessed, e.g., by the modularity. The algorithm is efficient in practice. Each relabeling iteration through the vertices has a computational complexity linear in the number of edges in the graph. The total number of iterations is not *a priori* clear, but relatively few iterations are needed to assign the final label to most of the vertices (over 95% of vertices in 5 iterations, see Refs. [4, 5]).

Two related works are of particular note. First, Tibély and Kertész [7] have identified the label propagation algorithm as formally equivalent to minimizing the Hamiltonian for a kinetic Potts model, and used this to argue

that, at least in some networks, the identified communities may be meaningless. Additionally, through empirical investigation of two real-world networks, Tibély and Kertész have shown that the number of distinct community solutions may be very large—much larger than the number of network vertices. Taken together, these observations highlight the need for further assessment of the quality of communities found using label propagation.

Second, Leung et al. [5] have examined the LPA as a basis for analyzing large networks, focusing on performance characteristics and limitations. They suggest a number of extensions and optimizations, resulting in a modified algorithm that is able to find communities in a network with tens of millions of edges in a few minutes using a desktop PC. This study thus suggests that label propagation has tremendous potential as an effective and efficient method for community identification.

III. AN OBJECTIVE FUNCTION FOR LABEL PROPAGATION

Thus far, the LPA has been presented operationally—the community solutions are defined as the outcome of a specific procedure. Alternatively, an equivalent mathematical formulation, first recognized by Tibély and Kertész [7], can be given, where community solutions are understood in terms of the results of applying an optimization procedure to an objective function. The optimization procedure is the LPA, while the objective function remains to be specified. The mathematical reformulation thus requires defining the objective function, which provides an alternate means of understanding solutions found by the LPA.

To effect this reformulation, we first express the LPA optimization procedure as

$$l'_v = \underset{l}{\operatorname{argmax}} \sum_{u \in \sigma(v)} \delta\left(l_u, l\right) \quad , \tag{4}$$

where l_u is the current label for vertex u, l'_v is the new label for vertex v, $\sigma\left(v\right)$ is the set of vertices neighboring v in the network, and δ is the Kronecker delta. In the event that multiple values would maximize the sum, the result of argmax_l should be taken as for the procedural description of LPA, i.e., keep the current label if it would satisfy Eq. (4), otherwise take a label at random that satisfies Eq. (4).

Equation (4) can be written in terms of the adjacency matrix $\mathbf{A}$ for the network, giving

$$l'_v = \underset{l}{\operatorname{argmax}} \sum_{u=1}^{n} A_{uv} \delta\left(l_u, l\right) \quad , \tag{5}$$

where n is the number of vertices in the network. Consistent with the LPA, the adjacency matrix elements A_{uv} are all elements of $\{0, 1\}$. However, the discrete nature of the A_{uv} is never made use of, so the form in Eq. (5) is equally applicable to weighted networks.

Next, we introduce an objective function H that is maximized by the optimization procedure. Intuitively, we can view the LPA as working to assign labels so as to increase the number of edges that connect vertices with identical labels. Formally, this number has the expression

$$H = \frac{1}{2} \sum_{v=1}^{n} \sum_{u \in \sigma(v)} \delta\left(l_u, l_v\right) \quad . \tag{6}$$

Equation (6) can be rewritten in terms of the network adjacency matrix, giving

$$H = \frac{1}{2} \sum_{v=1}^{n} \sum_{u=1}^{n} A_{uv} \delta\left(l_u, l_v\right) \quad . \tag{7}$$

We note that maximizing H is equivalent to minimizing the Hamiltonian for a ferromagnetic Potts model; this connection has been previously recognized by Tibély and Kertész [7]. The use of a Potts model Hamiltonian in network partitioning has been explored in depth by Reichardt and Bornholdt [10].

It remains to be verified that the optimization rule in Eq. (4) does in fact maximize the objective function in Eq. (7). Consider updating the label for some vertex x. We rewrite Eq. (7) to treat vertex x separately, yielding

$$H = \frac{1}{2} \left(\sum_{v \neq x} \sum_{u \neq x} A_{uv} \delta\left(l_u, l_v\right) + \sum_{u=1}^{n} A_{ux} \delta\left(l_u, l_x\right) + \sum_{v=1}^{n} A_{xv} \delta\left(l_x, l_v\right) - A_{xx} \right) \quad . \tag{8}$$

Taking advantage of the symmetry of the adjacency matrix, we can simplify Eq. (8), giving

$$H = \frac{1}{2}\left(\sum_{v\neq x}\sum_{u\neq x} A_{uv}\delta\left(l_u, l_v\right) - A_{xx}\right) + \sum_{u=1}^{n} A_{ux}\delta\left(l_u, l_x\right) \quad . \tag{9}$$

The final term on the right hand side of Eq. (9) is exactly of the form maximized by the LPA optimization rule as expressed in Eq. (5), while the other terms are independent of the label on vertex x. Thus, the objective function never decreases under the action of the LPA, ultimately reaching a local maximum or limit cycle.

An important property of the label propagation algorithm is immediately apparent from the form of H. For any network, the LPA allows an uninteresting trivial solution in which all vertices are assigned the same label [4]. From H, we see that the trivial solution is in fact the globally optimal solution. Other solutions found by label propagation correspond to local maxima of H.

As the LPA optimization procedure in Eq. (5) produces only local changes, the search for maxima in H is prone to becoming trapped at a local optimum instead of the global optimum. While normally a drawback of local search algorithms, this characteristic is essential to the function of the LPA: the trivial optimal solution is avoided by the dynamics of the local search algorithm, rather than through formal exclusion.

IV. DRAWBACKS OF LABEL PROPAGATION

The label propagation algorithm as a search scheme thus depends on a certain degree of ineffectuality. A typical way to attempt improvement of a local search algorithm is to make it more able to escape from local maxima in H. Such improvements to the LPA may be quite counterproductive, as better solutions in terms of H—notably, the global maximum—may be quite useless in practical terms. Despite this, label propagation in practice can produce communities that are of high quality in terms of, e.g., modularity: the local maxima are frustrated equilibria, with localized groups of well-connected vertices having the same label and with comparatively few edges between the groups.

Generally, there is a poor correspondence between H and our conceptual understanding of communities. Maximizing H, be it by label propagation or another approach, need not produce better communities. Regardless, using the LPA works by maximizing H, raising the question of whether, and in what sense, we are improving community quality. Operationally, it is again unclear what it might mean to try improving the LPA. Does improving the search efficacy actually give better communities? How do we prevent our optimizations from reaching the global maximum of H, or other uninteresting solutions with high values of H?

To illustrate the difficulties involved, we consider a possible optimization of the label propagation algorithm. When a vertex label is to be updated, it is necessary to handle the case where multiple labels are equally frequent for the neighboring vertices. In the standard LPA, these ties are broken by keeping the current label for the vertex, if it is one of the most frequent, or otherwise by selecting a label at random from the most frequent. In our optimized version, we will always select a label at random from the most frequent; in light of this additional randomization, we denote the modified algorithm as LPAr. The tie-breaking rule for the standard LPA corresponds to halting when a plateau in the H space is reached, while LPAr corresponds to allowing a random walk on the plateau in search of better solutions.

In Fig. 1, we show the number of communities found for one thousand applications of the standard LPA and the putatively optimized LPAr to networks derived from the Southern women data. The data were collected by Davis et al. [11] as part of an extensive study of class and race in the Deep South. The network represents interactions of a group of 18 women at 14 various events in and around Natchez, Mississippi during the 1930s. This much-studied network is typically found to have two communities using methods of social network analysis [12], in accord with the conclusions from the original ethnographic study. Unfortunately, our attempted optimization has a perverse result with the Southern women network: the principal effect of the optimization is to drastically increase the frequency at which the algorithm assigns the same label to all vertices, failing to capture any aspect of the known community structure.

At least in the Southern women network, several practical drawbacks arise from the key conceptual drawback discussed above. Optimization is made difficult, as seen in this case based on comparison with a known community structure. Further, the objective function optimized by LPA provides no mechanism for testing the quality of the resulting community solutions—we must instead assess quality through auxiliary considerations such as the number of communities or, e.g., the modularity Q of the community solution.

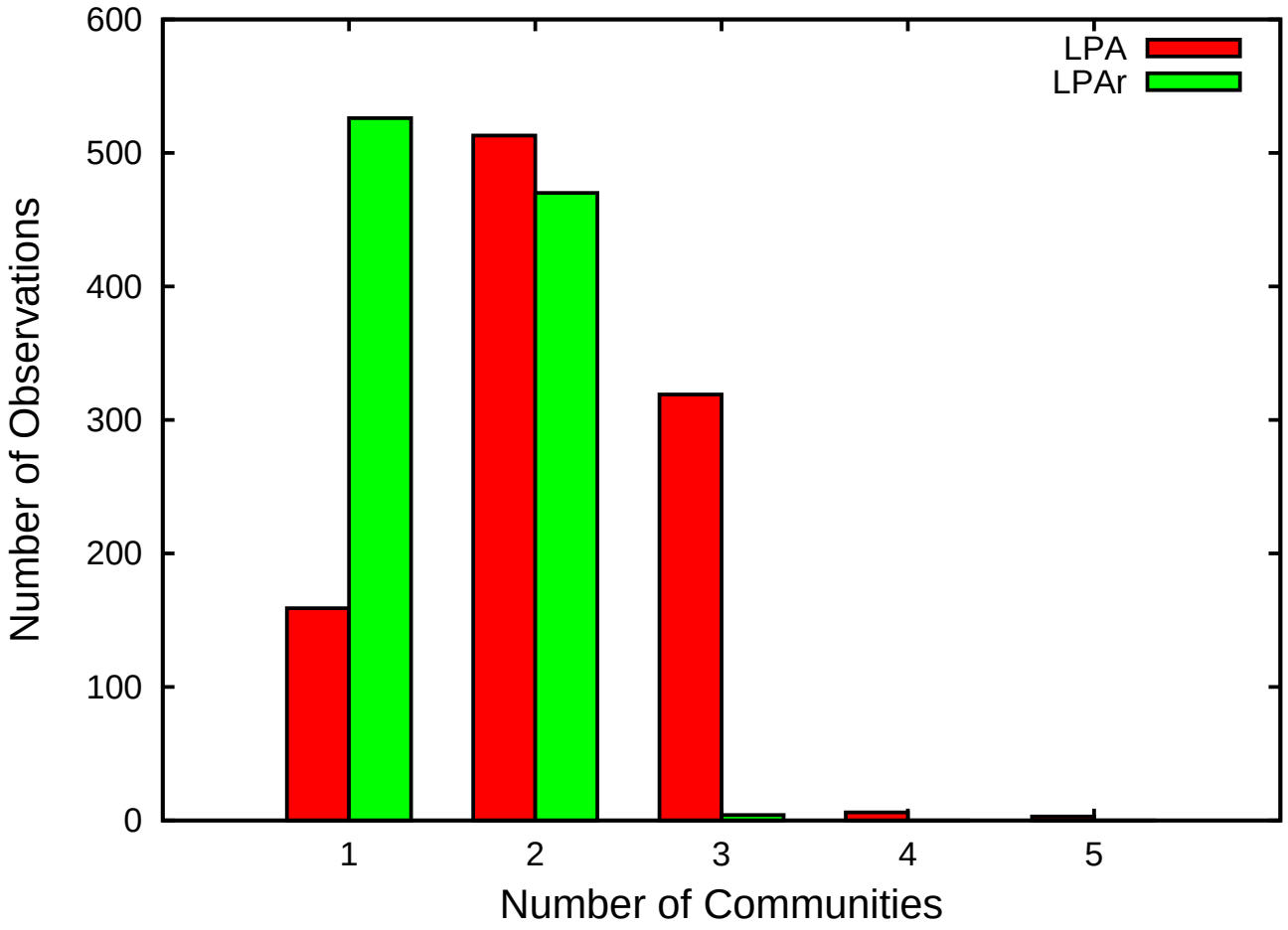

FIG. 1: An attempted optimization of the label propagation algorithm produces dubious gains for the Southern women network. The modified LPAr frequently produces the trivial solution, with all vertices assigned to the same community. In the network considered, we expect at least two communities based on the ethnographic study from which the data is drawn.

V. CONSTRAINED LABEL PROPAGATION

A well-established approach for eliminating undesirable solutions is to modify the objective function by adding a constraint term that penalizes the undesirable solutions. Denoting the modified objective function as H' and the penalty term as G, we have

$$H' = H - \lambda G \quad , \tag{10}$$

where λ is a parameter that weights G against the original objective function H. Numerous choices are possible for G; we consider three possibilities below.

Within the specific area of communication identification, the approach has been used at least since the landmark paper by Fu and Anderson [13] applying methods of statistical mechanics to combinatorial optimization problems, including graph bipartitioning. We base a first penalty term G_1 on their classic work. We seek to divide the vertices into groups of the same size. In terms of the labels, we define

$$\begin{aligned} G_1 &= \frac{1}{2}\sum_{l=1}^{n}\left(\sum_{v=1}^{n}\delta\left(l_v, l\right)\right)^2 \\ &= \frac{1}{2}\sum_{u=1}^{n}\sum_{v=1}^{n}\delta\left(l_v, l_u\right) \quad . \end{aligned} \tag{11}$$

The penalty term G_1 produces the smallest value when all vertices have unique labels, and the largest value when all vertices have the same label. Thus, the trivial global optimum of H is penalized and hopefully avoided.

Alternatively, following a strategy that mirrors contemporary methods for community detection, we can try to divide the vertices into groups which have a similar total degree. We define a second penalty term G_2 to capture this idea. The total degree K_l of the vertices with a given label l is

$$K_l = \sum_{i=1}^{n} k_i \delta\left(l_i, l\right) \quad , \tag{12}$$

where k_i is the degree of vertex i. A suitable definition for G_2 is

$$G_2 = \frac{1}{2} \sum_{l=1}^{n} K_l^2 \quad . \tag{13}$$

As with G_1, G_2 is minimal when all vertices have unique labels and maximal when all vertices have the same label, working to avoid the trivial global optimum.

We can rewrite G_2 in the form

$$\begin{aligned} G_2 &= \frac{1}{2} \sum_{l=1}^{n} \left(\sum_{v=1}^{n} k_v \delta\left(l_v, l\right) \right)^2 \\ &= \frac{1}{2} \sum_{u=1}^{n} \sum_{v=1}^{n} k_u k_v \delta\left(l_u, l_v\right) \quad . \end{aligned} \tag{14}$$

Incorporating G_2 into H', we obtain

$$H' = \frac{1}{2} \sum_{u=1}^{n} \sum_{v=1}^{n} \left(A_{uv} - \lambda k_u k_v\right) \delta\left(l_u, l_v\right) \quad . \tag{15}$$

If we select

$$\lambda = \frac{1}{2m} \quad , \tag{16}$$

where m is the number of edges in the network, the objective function may be written as

$$H' = mQ \quad . \tag{17}$$

In Eq. (17), Q is the standard modularity measure [6].

Recalling that the label propagation rule as given by Eq. (5) requires only that a symmetric matrix be used, we can see from Eq. (15) that modularity can be be locally maximized by the label propagation algorithm; we denote this modularity-specialized algorithm as LPAm. Implementation issues are described in appendix A. We note that LPAm, due to the effect of G_2, is well suited to aggressive optimization, but we do not pursue such optimizations in the present work.

The penalty term G_2 plays the same role as the null model network used to define the modularity (see, e.g., Ref. [6]). The idea holds quite generally: various null model networks could be used to define specialized modularity measures, or penalty terms could equivalently be introduced into the objective function. This allows the interesting historical interpretation that Fu and Anderson [13] made use of a modularity measure for community identification over two decades ago.

As a further example, we develop an analogous label propagation algorithm to maximize a recently introduced [14] version of modularity adapted to the important special class of bipartite networks. The vertices of a bipartite network can be partitioned into two disjoint sets such that no two vertices within the same set are adjacent; equivalently, the vertices in a bipartite graph can be assigned one of two colors, say red and blue, with no edges present between vertices bearing the same color. There are thus two distinct kinds of vertices, providing a natural representation for many affiliation or interaction networks, with one kind of vertex representing actors and the other representing relations.

The distinction between the two parts of the network can be incorporated into a modularity measure by defining a suitable null network model. In contrast to the standard choice given in Eq. (2), the two kinds of vertices must be treated separately, with non-zero probability of an edge only between vertices belonging to different parts of the network. For a red vertex i with degree k_i and a blue vertex j with degree d_j, the null model is defined so that

$$P_{ij} = \frac{k_i d_j}{m} \quad . \tag{18}$$

Using Eq. (18), the bipartite modularity $Q^{\mathcal{B}}$ is

$$Q^{\mathcal{B}} = \frac{1}{m} \sum_{i,j}{}' \left(A_{ij} - \frac{k_i d_j}{m} \right) \delta\left(g_i, g_j\right) \quad . \tag{19}$$

The sums in Eq. (19) are to be interpreted as running over the vertices in the two parts of the network, i.e., i is restricted to run over only the red vertices, while j is restricted to run over only the blue vertices.

For the present work, it is simpler to allow unrestricted sums over all the vertices. To do this, for each vertex v we associate two degree measures, a red degree k_v and a blue degree d_v. If vertex v is red, we require $d_v = 0$, while if it is blue we require $k_v = 0$. In either case, the non-zero degree is the number of edges incident on the vertex. With this construction, Eq. (19) becomes

$$Q^{\mathcal{B}} = \frac{1}{2m} \sum_{i=1}^{n} \sum_{j=1}^{n} \left(A_{ij} - \frac{2k_i d_j}{m} \right) \delta\left(g_i, g_j\right) \quad , \tag{20}$$

where now the sums run over all vertices.

We now define a penalty term G_3 for bipartite networks as

$$G_3 = \frac{1}{2} \sum_{l=1}^{n} K_l D_l \quad , \tag{21}$$

where

$$K_l = \sum_{u=1}^{n} k_u \delta\left(l_u, l\right) \quad , \tag{22}$$

$$D_l = \sum_{u=1}^{n} d_u \delta\left(l_u, l\right) \quad . \tag{23}$$

Equations (21) through (23) adapt Eqs. (12) and (13) to bipartite networks.

We can rewrite G_3 as

$$\begin{aligned} G_3 &= \frac{1}{2} \sum_{l=1}^{n} \left(\sum_{u=1}^{n} k_u \delta\left(l_u, l\right) \sum_{v=1}^{n} d_v \delta\left(l_v, l\right) \right) \\ &= \frac{1}{2} \sum_{u=1}^{n} \sum_{v=1}^{n} k_u d_v \delta\left(l_u, l_v\right) \quad . \end{aligned} \tag{24}$$

Writing the full objective function, we have

$$H' = \frac{1}{2} \sum_{u=1}^{n} \sum_{v=1}^{n} \left(A_{uv} - \lambda k_u d_v \right) \delta\left(l_u, l_v\right) \quad . \tag{25}$$

With

$$\lambda = \frac{2}{m} \quad , \tag{26}$$

Eq. (25) becomes

$$H' = m Q^{\mathcal{B}} \quad . \tag{27}$$

The label propagation rule can again be used to maximize the bipartite modularity; we denote the algorithm as LPAb. Implementation issues for LPAb are treated in appendix B.

VI. APPLICATIONS AND PERFORMANCE

A. Unipartite networks

We now turn to a comparison of the quality of solutions found by the various label propagation algorithms discussed above. To quantify the solution quality, we will focus principally on the modularity Q, although it is not strictly optimized except by LPAm. Along with the LPA, LPAr, and LPAm variants discussed above, we will additionally consider a hybrid algorithm, consisting of the standard LPA followed by optimization with LPAm. The hybrid approach ensures that we are at a maximum in the modularity, rather than just finding a solution that hopefully offers a high value of Q.

To begin, we apply the algorithms to randomly generated networks with a known community structure. The most typical such class of networks, introduced by Girvan and Newman [15], consists of four communities, each containing 32 vertices. Edges exist between pairs of vertices belonging to the same community with probability p_{in} and between all other pairs of vertices with probability p_{out}. The probabilities p_{in} and p_{out} are set so as to preserve the average degree $\langle k \rangle$ of the vertices at a value of 16, while varying the average number of edges z_{out} between a vertex and members of other communities. As z_{out} increases, the communities become increasingly difficult to identify. Although these model networks differ significantly from real networks with community structure [16], they do provide a simple initial test of community detection algorithms.

In Fig. 2, we show the modularity values for communities found by the four algorithms. Each point shown gives the average modularity from communities found in 1000 instances of the random network model. As expected, Q drops as z_{out} increases.

Since we know the actual communities for the model networks, we may additionally assess the accuracy of the label assignments by directly comparing to the known values. We use the normalized mutual information I_{norm} [2] for the comparison. Consider two schemes X and Y for dividing the n vertices into community groups. The probability $P(X = x, Y = y)$ that a vertex is assigned to community x in scheme X and to community y in scheme Y is taken to be proportional to the size of the intersection between the sets of vertices C_x and C_y constituting the communities, so that

$$P(X = x, Y = y) = \frac{|C_x \cap C_y|}{n} \quad . \tag{28}$$

Using the probability as defined in Eq. (28), we can calculate the normalized mutual information as

$$I_{\text{norm}}(X, Y) = \frac{2I(X, Y)}{H(X) + H(Y)} \quad . \tag{29}$$

Equation (29) is expressed in terms of the usual mutual information $I(X, Y)$ and entropies $H(X)$ and $H(X)$ [17], defined as

$$I(X, Y) = \sum_{x,y} P(X, Y) \log \frac{P(X, Y)}{P(X) P(Y)} \tag{30}$$

$$H(X) = -\sum_{x} P(X) \log P(X) \tag{31}$$

$$H(Y) = -\sum_{y} P(Y) \log P(Y) \quad . \tag{32}$$

In Eqs. (29) through (32), we have made use of the common shorthand abbreviations $P(X = x, Y = y) = P(X, Y)$, $P(X = x) = P(X)$, and $P(Y = y) = P(Y)$. The base of the logarithms in Eqs. (30) through (32) is arbitrary, as the computed measures only appear in the ratio in Eq. (29).

The normalized mutual information allows us to measure the amount of information common to two different partitioning schemes. Accordingly, we can explore the efficacy of the algorithm by taking one of the partitions to be the known modular structure of the model networks and the other to be the structure found using label propagation. When the found modules match the real ones, we have $I_{\text{norm}} = 1$, and when they are independent of the real ones, we have $I_{\text{norm}} = 0$. Thus, as z_{out} increases, we expect I_{norm} to decrease. In Fig. 3, we present values of I_{norm} from comparison of the real communities to the same community solutions used for the Q calculations in Fig. 2, observing the expected decrease from $I_{\text{norm}} = 1$ to $I_{\text{norm}} = 0$.

From Figs. 2 and 3, it is tempting to conclude that LPAm is superior to the other label propagation variants. However, this conclusion is not borne out when the algorithms are applied to real networks. In Table I, we list several

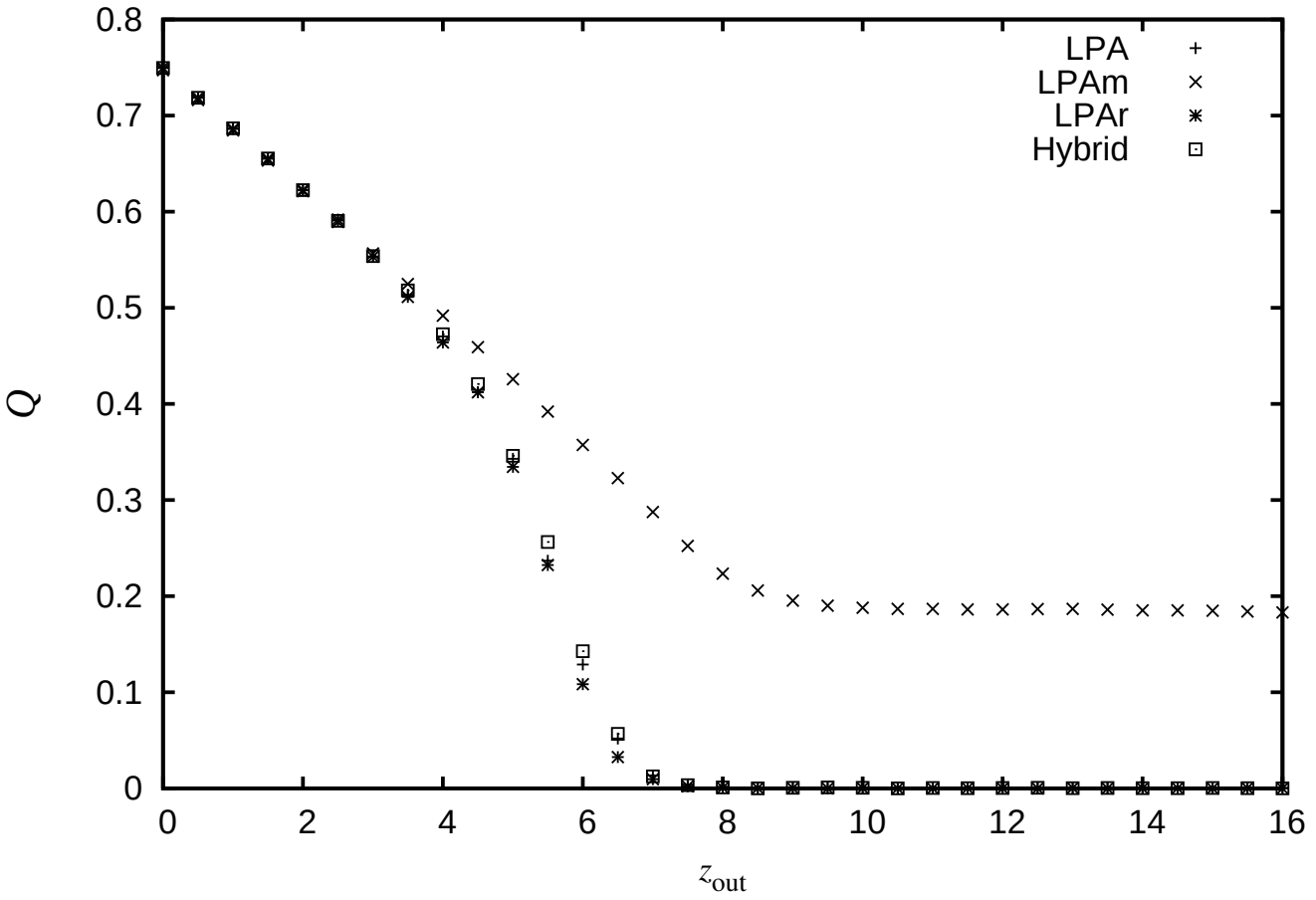

FIG. 2: Modularity Q of community solutions from random networks with known community structures. Each point shows the average Q over 1000 instances of the random networks in relation to the average number z_{out} of inter-community links for each vertex. The hybrid algorithm consists of allowing the standard LPA to run its course and find a solution, followed by application of LPAm to the LPA solution in order to ensure that a local maximum of Q is reached. Error bars are smaller than the points.

networks that we have investigated using the label propagation algorithms. The networks considered are a network of friendships between members of a university karate club [18]; a network of frequent associations between dolphins living near Doubtful Sound, New Zealand [19]; a network of collaborations between jazz musicians [20]; a network of co-authorships for scientific papers concerning networks [21]; and a network of co-authorships for scientific preprints posted to the condensed matter archive [22] between the years 1995 and 2003 [8]. We give their sizes in terms of the number of vertices n and number of edges m. To indicate the degree to which the networks feature community structures, we also provide the modularity Q, as determined using a greedy agglomerative hierarchical (GAH) method based on that of Clauset et al. [9], wherein pairs of communities are successively merged so as to cause the largest possible increase in Q at each step. While edge weights are available in some cases, in this work we uniformly treat all network edges as unweighted.

For each of the networks, we identify communities using each of the algorithms LPA, LPAm, and LPAr. Additionally, we consider a hybrid algorithm consisting of LPA followed by LPAm, thus ensuring that we are at a maximum of the Q. We applied each of the four algorithms one hundred times to each of the networks. In Table II, we show the maximum modularity found in the samples, suggesting the potential performance, while in Table III, we show the mean modularity, revealing the expected performance. From the tables, we can see that no algorithm variant is clearly superior, suggesting that the four variants all explore slightly different portions of the solution space. Interestingly, the LPAr variant, which worked poorly when applied to the Southern women network (section IV), provides the best results on the two large co-authorship networks. We note that the label propagation variants produce community solutions with modularity values similar to those found with the GAH approach and shown in Table I.

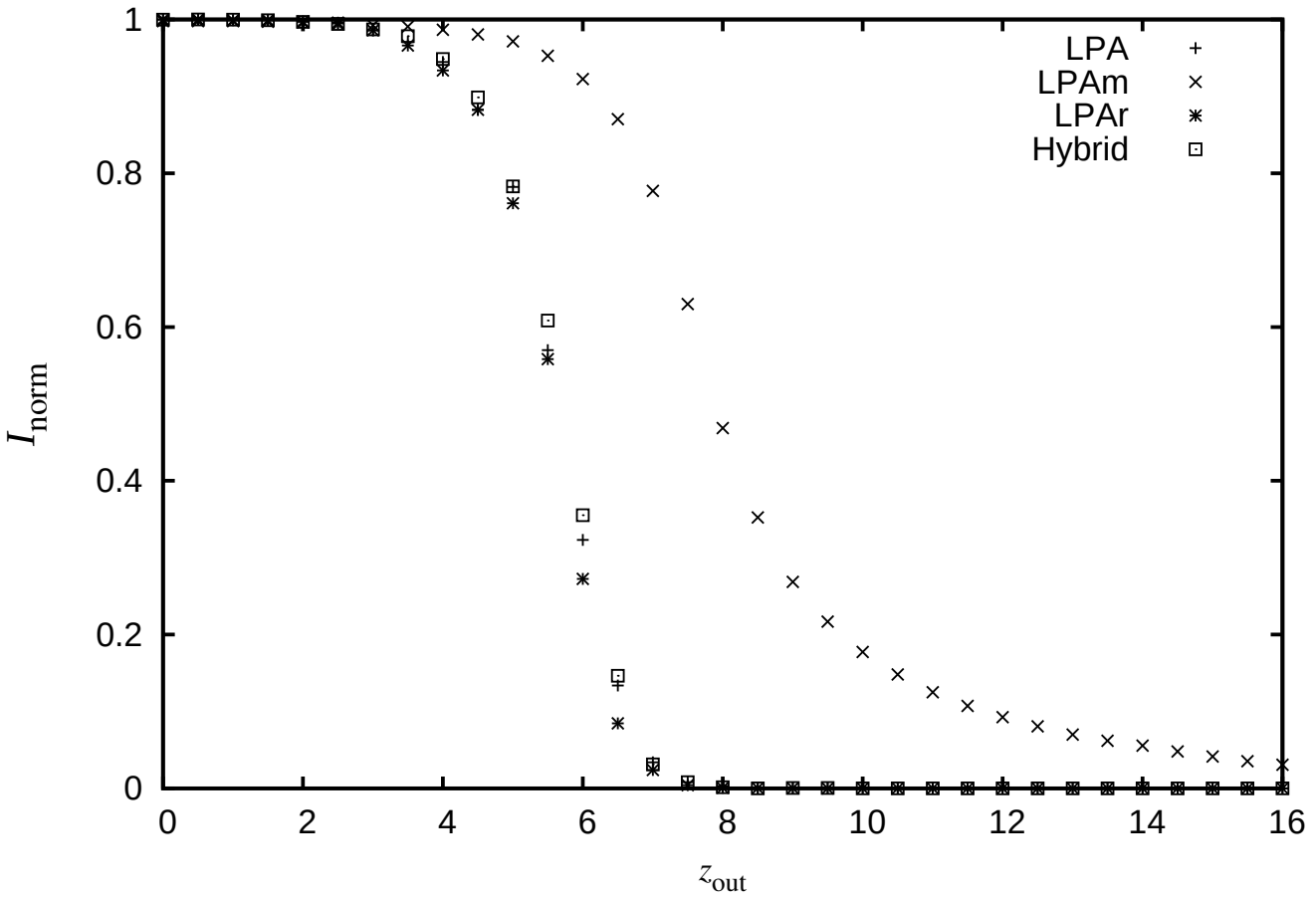

FIG. 3: Accuracy of community solutions from random networks with known community structures. Accuracy is quantified by the normalized mutual information I_{norm} between the found and actual community solutions. Each point shows the normalized mutual information I_{norm} over 1000 instances of the random networks in relation to the average number z_{out} of inter-community links for each vertex. The hybrid algorithm consists of allowing the standard LPA to run its course and find a solution, followed by application of LPAm to the LPA solution in order to ensure that a local maximum of Q is reached. Error bars are smaller than the points.

Network	n	m	Q
karate	34	78	0.3807
dolphins	62	159	0.4923
jazz	198	2742	0.4389
network science	1589	2742	0.9555
condmat 2003	31163	120029	0.6885

TABLE I: Basic properties of networks used to test label propagation algorithm variants. The sizes of the network are described by the number of vertices n and number of edges m. Each network has significant modular character, as indicated by the modularity Q.

B. Bipartite networks

As we did above for unipartite networks, we next quantify the quality of community solutions found in bipartite networks. We measure community quality using the bipartite modularity $Q^{\mathcal{B}}$, calculating values for the LPA, LPAr, and LPAb variants. Again, we consider a hybrid algorithm, consisting of LPA followed by LPAb, ensuring that the solutions are at maxima in $Q^{\mathcal{B}}$.

Network	LPA	LPAm	LPAr	Hybrid
karate	0.4156	0.4000	0.4156	0.4198
dolphins	0.5237	0.5157	0.5265	0.5253
jazz	0.4424	0.4448	0.4428	0.4442
network science	0.8924	0.8723	0.9163	0.8934
condmat 2003	0.6228	0.5947	0.6578	0.6360

TABLE II: Maximum modularity Q found for network community assignments. Values were calculated using one hundred samples for each network for each of the standard LPA, LPAm, LPAr, and a hybrid approach consisting of maximization with LPA followed by maximization with LPAm.

Network	LPA	LPAm	LPAr	Hybrid
karate	0.366(6)	0.347(3)	0.352(9)	0.386(4)
dolphins	0.484(4)	0.4956(8)	0.484(5)	0.495(3)
jazz	0.336(9)	0.4351(9)	0.34(1)	0.366(7)
network science	0.8792(6)	0.8618(5)	0.9046(5)	0.8806(6)
condmat 2003	0.6073(6)	0.5828(4)	0.6420(6)	0.6139(9)

TABLE III: Mean modularity Q found for network community assignments. Values were calculated using one hundred samples for each network for each of the standard LPA, LPAm, LPAr, and a hybrid approach consisting of maximization with LPA followed by maximization with LPAm. The uncertainty of the final digit, calculated as the standard error of the mean, is shown parenthetically.

We examine the performance using four real-world bipartite networks. The networks are the Southern women network, described above in section IV; a network describing corporate interlocks in Scotland, based on the membership of boards of directors for Scottish firms during 1904–5 [23]; and bipartite versions of the condensed matter and network science co-authorship networks considered in section VI A, including authors and their papers as the two parts of the network. In Table IV, we indicate the size and extent of community structure in the networks. We show the size using the number of vertices p and q in the two parts of the networks, as well as the number of edges m. We show the extent of community structure using the bipartite modularity $Q^{\mathcal{B}}$, as determined using a greedy agglomerative hierarchical method, analogous to that commonly used for unipartite networks [8, 9].

To each network, we apply each label propagation algorithm one hundred times. The maximum and mean values found for $Q^{\mathcal{B}}$ are given in Tables V and VI, respectively. For the Southern women network, we note that LPAr is clearly the worst of the algorithms considered, consistent with its tendency to assign the same label to all vertices, as seen in Fig. 1. Further, the improved performance of LPAb on the Southern women network in terms of the average $Q^{\mathcal{B}}$ indicates that the inclusion of G_3 reduces the frequent appearance of the trivial solution with all vertices in the same community.

Despite the success of LPAb on the Southern women network, it is less successful on the other networks. Performance is quite similar for LPA and LPAb on the Scotland corporate interlocks network, but LPAb is otherwise outperformed by the other label propagation variants. Indeed, LPAr provides the best results for the larger networks, in contrast to its poor results for the Southern women network. Values of $Q^{\mathcal{B}}$ for community solutions found using the label propagation variants are generally somewhat less than the values, shown in Table IV, for communities found using a

Network	p	q	m	$Q^{\mathcal{B}}$
Southern women	14	18	89	0.3430
Scotland interlocks	108	136	358	0.6969
network science	959	1588	2580	0.9695
condmat 2003	31162	47055	134600	0.8700

TABLE IV: Basic properties of bipartite networks used to test label propagation algorithm variants. The sizes of the network are described by the numbers of vertices p and q in the two parts of the network and by the number of edges m. Each network has significant modular character, as indicated by the bipartite modularity $Q^{\mathcal{B}}$.

Network	LPA	LPAb	LPAr	Hybrid
Southern women	0.3212	0.3192	0.3184	0.3257
Scotland interlocks	0.5782	0.5783	0.6552	0.5975
network science	0.8137	0.7807	0.8948	0.8172
condmat 2003	0.6378	0.6179	0.7232	0.6587

TABLE V: Maximum bipartite modularity $Q^{\mathcal{B}}$ found for bipartite network community assignments. Values were calculated using one hundred samples for each network for each of the standard LPA, LPAb, LPAr, and a hybrid approach consisting of maximization with LPA followed by maximization with LPAb.

Network	LPA	LPAb	LPAr	Hybrid
Southern women	0.19(1)	0.250(3)	0.17(1)	0.27(1)
Scotland interlocks	0.543(1)	0.548(2)	0.633(1)	0.568(1)
network science	0.788(1)	0.7624(6)	0.8733(8)	0.7986(8)
condmat 2003	0.6314(3)	0.6142(1)	0.7183(2)	0.6536(2)

TABLE VI: Mean bipartite modularity $Q^{\mathcal{B}}$ found for bipartite network community assignments. Values were calculated using one hundred samples for each network for each of the standard LPA, LPAb, LPAr, and a hybrid approach consisting of maximization with LPA followed by maximization with LPAb.

greedy agglomerative hierarchical approach.

VII. DISCUSSION

We have examined the label-propagation algorithm as an optimization problem, identifying community solutions that it finds with the maxima of an objective function. The objective function, which is just the number of network edges connecting vertices with the same labels, has the significant conceptual drawback that increasing the objective function need not produce what we would consider to be better communities. Markedly, the globally optimal solution is completely uninformative, with all vertices in the same community. Label propagation thus depends on reaching one of the large number of local maxima in the objective function to avoid the trivial global solution. Attempts to improve on the algorithm may be counterproductive, giving less information while reaching nominally better solutions. By modifying the objective function, we defined several label-propagation algorithms that are constrained to avoid assigning all vertices to the same community. One of the constrained label-propagation algorithms, LPAm, finds local maxima in the modularity Q; another, LPAb, finds local maxima in a modified modularity $Q^{\mathcal{B}}$ for bipartite networks.

Although formally equivalent, there are important conceptual differences between the usual definition of the modularity Q in terms of a null model network and the version based on constraints presented here. For example, the parameter λ seems quite arbitrarily chosen in the constraint-based version. In fact, the community solutions found by LPAm are not especially sensitive to the choice of λ. The value can, for instance, be cut in half to $\lambda = 1/4m$ with significant change only in the case of the mean modularity for Zachary's karate network—in which the mean modularity value actually increases by about 10%.

More significantly, the constraint as given in Eq. (13) makes clear that modularity favors communities of similar size, with size measured by the total degree of the vertices in the community. As the distribution of community sizes may be far from uniform (see, for example, Fig. 3 in Ref. [9]), the constraint approach points immediately towards a practical difficulty in detecting community by maximizing modularity. In contrast, difficulties due to varying community sizes were recognized [24] only some time after the original introduction of modularity using a null model.

Corresponding properties hold in the case of the bipartite modularity $Q^{\mathcal{B}}$. Again, λ seems arbitrarily chosen; halving the parameter value to $\lambda = 1/m$ again only causes a significant change for the small Southern women network, increasing the mean bipartite modularity found by about 10%. In the bipartite case, communities of similar size are also favored, but the relevant size is now the geometric mean of the total degrees within the community for the two parts of the network, as seen in Eq. (21). We thus expect that community identification methods based on maximizing $Q^{\mathcal{B}}$ will also have difficulties with networks consisting of communities of diverse sizes. Although this latter fact has been anticipated [14] based on parallels to the unipartite case, it has not been previously demonstrated.

In light of the results for the real-world networks (Tables II and III for unipartite networks, Tables V and VI for

bipartite networks), it seems clear that the main label propagation variants we have considered—LPA, LPAm, LPAr, LPAb—all give good community results. The performance differences indicate that the algorithm variants explore slightly different portions of the community solution space. No variant is clearly superior, which is not surprising given that we are trying to identify communities without prior information on their number, size, or nature.

When compared to the modularity values for community solutions generated by greedy agglomerative hierarchical methods, the label propagation variants appear to provide no advantage or, in the case of bipartite networks, to entail a distinct disadvantage. We stress that the difference in modularity values should not be overvalued, for two main reasons. First, the modularity measure, while popular, is not the only possibility, nor is it without drawbacks (see, e.g., Ref. [25]). Second, the algorithms are quite different, so no single point of comparison will be determinative in general. A more thorough characterization of performance is needed to establish reliable guidelines for choosing appropriate algorithms to analyze particular networks; this will be the subject of future work.

The performance of LPAm is especially interesting: although it is the only variant directly maximizing Q, other variants produce better results in terms of Q for some of the networks considered. This appears to be due to a fundamental difference in the role played by the modularity in the algorithm variants. Lacking an objective function, Raghavan et al. [4] used the modularity of the final community solution to assess the acceptability of their LPA, as did we when assessing LPAr in the present work. Thus, in LPA and LPAr, the modularity is used diagnostically to select a best result from candidate solutions produced based on other considerations. In contrast, the modularity plays an essential role in LPAm, impacting the final community solution as well as the intermediate community states reached during the course of the algorithm. The dynamical path followed through the space of label assignments is driven to favor states where all communities are similar in total degree, although there is little reason to believe such paths are universally ideal or particularly free of local maxima. Thus, the null model network used in defining the modularity—regardless of its suitability as a model of the final communities—may be an impractical model of the intermediate communities. This might be addressed by varying G, gradually introducing the penalty term G_2 and thus the null network mode. Similar considerations hold for LPAb and the corresponding null network models for bipartite networks.

Overall, we have found the label propagation algorithm to be a promising approach to understanding networks, with a number of desirable qualities. Label propagation seems well suited as a basis for more specialized community-detection methods, as well as application to other aspects of networks besides community structure. A clear understanding of the drawbacks of label propagation, as well as its strengths, will help to avoid problems and facilitate further applications.

Acknowledgments

We thank Mark Newman for providing the bipartite versions of networks describing co-authorships in condensed matter and in network sciences. This work has been supported by the European FP6-NEST-Adventure Programme, contract number 028875.

[1] M. E. J. Newman, Eur. Phys. J. B **38**, 321 (2004), URL http://www-personal.umich.edu/~mejn/papers/epjb.pdf.
[2] L. Danon, A. Díaz-Guilera, J. Duch, and A. Arenas, J. Stat. Mech. p. P09008 (2005), URL http://www.iop.org/EJ/article/1742-5468/2005/09/P09008/jstat5_09_p09008.html.
[3] S. Fortunato and C. Castellano, in *Encyclopedia of Complexity and System Science* (Springer, 2008).
[4] U. N. Raghavan, R. Albert, and S. Kumara, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) **76**, 036106 (pages 11) (2007), URL http://link.aps.org/abstract/PRE/v76/e036106.
[5] I. X. Y. Leung, P. Hui, P. Liò, and J. Crowcroft, *Towards real time community detection in large networks* (2008), arXiv:0808.2633v3.
[6] M. E. J. Newman and M. Girvan, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) **69**, 026113 (pages 15) (2004), URL http://link.aps.org/abstract/PRE/v69/e026113.
[7] G. Tibély and J. Kertész, Physica A: Statistical Mechanics and its Applications **387**, 4982 (2008).
[8] M. E. J. Newman, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) **69**, 066133 (2004), URL http://link.aps.org/abstract/PRE/v69/e066133.
[9] A. Clauset, M. E. J. Newman, and C. Moore, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) **70**, 066111 (pages 6) (2004), URL http://link.aps.org/abstract/PRE/v70/e066111.
[10] J. Reichardt and S. Bornholdt, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) **74**, 016110 (pages 14) (2006), cond-mat/0603718, URL http://link.aps.org/abstract/PRE/v74/e016110.
[11] A. Davis, B. B. Gardner, and M. R. Gardner, *Deep South* (University of Chicago Press, 1941).

[12] L. Freeman, in *Dynamic Social Network Modeling and Analysis*, edited by R. Breiger, K. Carley, and P. Pattison (The National Academies Press, Washington, DC, 2003), URL http://moreno.ss.uci.edu/85.pdf.
[13] Y. Fu and P. W. Anderson, Journal of Physics A: Mathematical and General **19**, 1605 (1986), URL http://stacks.iop.org/0305-4470/19/1605.
[14] M. J. Barber, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) **76**, 066102 (2007).
[15] M. Girvan and M. E. J. Newman, PNAS **99**, 7821 (2002), URL http://www.pnas.org/cgi/content/abstract/99/12/7821.
[16] A. Lancichinetti, S. Fortunato, and F. Radicchi, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) **78**, 046110 (pages 5) (2008), URL http://link.aps.org/abstract/PRE/v78/e046110.
[17] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley Series in Telecommunications (Wiley-interscience, New York, NY, 1991).
[18] W. W. Zachary, Journal of Anthropological Research **33**, 452 (1977).
[19] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson, Behavioral Ecology and Sociobiology **54**, 396 (2003).
[20] P. M. Gleiser and L. Danon, Advances in Complex Systems **6**, 565 (2003).
[21] M. E. J. Newman, Physical Review E (Statistical, Nonlinear, and Soft Matter Physics) **74**, 036104 (2006), arXiv:physics/0605087v3, URL http://link.aps.org/abstract/PRE/v74/e036104.
[22] URL http://arxiv.org/archive/cond-mat.
[23] J. Scott and M. Hughes, *The anatomy of Scottish capital: Scottish companies and Scottish capital, 1900–1979* (Croom Helm, London, 1980).
[24] L. Danon, A. Diaz-Guilera, and A. Arenas, Journal of Statistical Mechanics: Theory and Experiment **2006**, P11010 (2006), URL http://stacks.iop.org/1742-5468/2006/P11010.
[25] S. Fortunato and M. Barthelemy, PNAS **104**, 36 (2007), URL http://www.pnas.org/cgi/reprint/104/1/36.pdf.

APPENDIX A: A LABEL-PROPAGATION ALGORITHM FOR MAXIMIZING MODULARITY

The label propagation algorithm presented by Raghavan et al. [4] has desirable performance properties. Each relabeling iteration through the vertices has a computational (time) complexity $O(m)$ linear in the number of edges m in the graph. For many networks, the number of vertices n scales with the number of edges, so the computational complexity for each relabeling step can instead be given as $O(n)$.

As seen in section V, the objective function for the LPA can be constrained to reproduce the modularity. Consequently, it is necessary to adapt the algorithm itself to obtain an efficient procedure for maximizing the modularity. Modifications can be made so as to maintain the $O(m)$ time complexity. Here, we consider the constraint G_2 given in Eq. (14), i.e., we implement LPAm.

First, consider the objective function from Eq. (7). Recall that the LPA update rule (Eq. (5)) can be applied with any symmetric matrix B_{uv} playing the role of the adjacency matrix A_{uv} (see section III). Further, it is clear that the objective function may be shifted by adding an arbitrary constant C without altering the locations of the maxima in the space of label assignments. By setting $C = -\sum_{u=1}^{n} B_{uu}$, we eliminate the diagonal elements B_{uu} from consideration, producing an objective function

$$H = \sum_{v=1}^{n} \sum_{u \neq v} B_{uv} \delta(l_u, l_v) \tag{A1}$$

and update rule

$$l'_v = \underset{l}{\operatorname{argmax}} \sum_{u \neq v} B_{uv} \delta(l_u, l) \quad . \tag{A2}$$

The above transformation eliminates constant self-interaction terms.

Next, identify B_{uv} as $A_{uv} - \lambda k_u k_v$ to match the LPAm variant, giving

$$l'_v = \underset{l}{\operatorname{argmax}} \sum_{u \neq v} (A_{uv} - \lambda k_u k_v) \delta(l_u, l) \tag{A3}$$

or, equivalently,

$$l'_v = \underset{l}{\operatorname{argmax}} \left(\sum_{u \neq v} A_{uv} \delta(l_u, l) - \lambda k_v \sum_{u \neq v} k_u \delta(l_u, l) \right) \quad . \tag{A4}$$

The first sum in Eq. (A4) corresponds to the counting of labels on neighboring vertices in the original label propagation algorithm. Write this as

$$N_{vl} = \sum_{u \neq v} A_{uv} \delta\left(l_u, l\right) \quad . \tag{A5}$$

The second sum in Eq. (A4) can be rewritten as

$$\sum_{u \neq v} k_u \delta\left(l_u, l\right) = K_l - k_v \delta\left(l_v, l\right) \quad , \tag{A6}$$

where

$$K_l = \sum_{u=1}^{n} k_u \delta\left(l_u, l\right) \quad . \tag{A7}$$

Analogously to the volume of a graph, K_l can be viewed as a sort of volume for the labels.

Incorporating Eqs. (A5) and (A7) into Eq. (A4), we obtain

$$l'_v = \underset{l}{\operatorname{argmax}} \left(N_{vl} - \lambda k_v K_l + \lambda k_v^2 \delta\left(l_v, l\right)\right) \quad . \tag{A8}$$

The modified label propagation rule, as expressed in Eq. (A8), can be readily implemented so that each pass through the vertices requires $O\left(m\right)$ worst-case time complexity.

The algorithm is initialized by assigning a unique numerical label l to each vertex and by setting K_l to the degree of the vertex. The first term, N_{vl}, requires that the labels of the neighbors for each vertex be counted and is thus $O\left(m\right)$; this is unsurprising as it is equivalent to the unmodified label propagation algorithm, which is $O\left(m\right)$. The second term appears to require that each possible label be checked for each vertex, giving $O\left(n^2\right)$. However, it is only necessary to consider the labels of the neighbors for each vertex—no other label can make a positive contribution to the modularity, but a zero contribution can be had by assigning an unused label. A list of unused labels can be kept, allowing $O\left(1\right)$ access. Additionally, the K_l must be updated if the label changes, but this is also $O\left(1\right)$ for each vertex. In total, checking and updating the K_l terms for all vertices is $O\left(m\right)$. The final term in Eq. (A8) is $O\left(n\right)$ in total. With all three terms taken into account, the modified algorithm thus has worst-case $O\left(m\right)$ time complexity.

APPENDIX B: A LABEL-PROPAGATION ALGORITHM FOR MAXIMIZING BIPARTITE MODULARITY

In Eq. (25), we have presented an objective function corresponding to the bipartite modularity $Q^{\mathcal{B}}$, with form

$$H' = \frac{1}{2} \sum_{u=1}^{n} \sum_{v=1}^{n} \left(A_{uv} - \lambda k_u d_v\right) \delta\left(l_u, l_v\right) \quad . \tag{B1}$$

We cannot directly apply the label propagation update rule from Eq. (5), as $A_{uv} - \lambda k_u d_v$ is in general asymmetric. Despite this, we can define a label propagation rule for H'.

We rewrite Eq. (25) by first taking advantage of the symmetry of A_{uv} and $\delta\left(l_u, l_v\right)$, giving

$$H' = \frac{1}{2} \sum_{u=1}^{n} \sum_{v=1}^{n} \left(A_{vu} - \lambda k_u d_v\right) \delta\left(l_v, l_u\right) \quad . \tag{B2}$$

Next, we switch the dummy indices u and v, resulting in

$$H' = \frac{1}{2} \sum_{u=1}^{n} \sum_{v=1}^{n} \left(A_{uv} - \lambda k_v d_u\right) \delta\left(l_u, l_v\right) \quad . \tag{B3}$$

Averaging Eqs. (B1) and (B3), we obtain

$$H' = \frac{1}{2} \sum_{u=1}^{n} \sum_{v=1}^{n} \left(A_{uv} - \frac{\lambda}{2} \left(k_u d_v + k_v d_u\right)\right) \delta\left(l_u, l_v\right) \quad , \tag{B4}$$

which is in terms of a symmetric matrix and thus suitable for use with Eq. (5).

The objective function, as expressed in Eq. (B4), can be converted into the LPAb label propagation rule for bipartite modularity in a fashion directly parallel to that presented in appendix A. The resulting update rule has the form

$$l'_v = \underset{l}{\operatorname{argmax}} \left(N_{vl} - \frac{\lambda d_v}{2} K_l - \frac{\lambda k_v}{2} D_l + \frac{\lambda}{2} k_v^2 \delta\left(l_v, l\right) + \frac{\lambda}{2} d_v^2 \delta\left(l_v, l\right) \right) \quad , \tag{B5}$$

where

$$K_l = \sum_{u=1}^{n} k_u \delta\left(l_u, l\right) \quad , \tag{B6}$$

$$D_l = \sum_{u=1}^{n} d_u \delta\left(l_u, l\right) \quad . \tag{B7}$$

By updating K_l and D_l when labels change, the algorithm can be implemented efficiently. The details, omitted here, are similar to those given in appendix A and result in the same $O\left(m\right)$ worst-case time complexity for each iteration of LPAb.